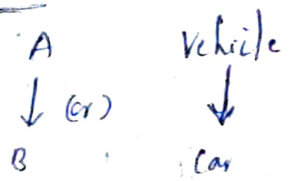
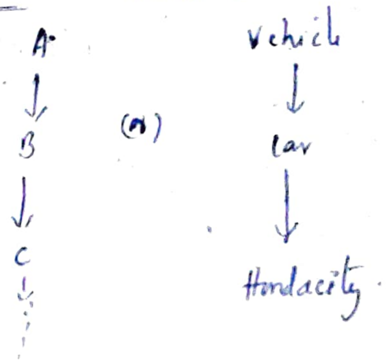
5. Inheritance Types

Types of Inheritance

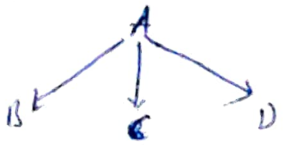
1) Single inheritance.

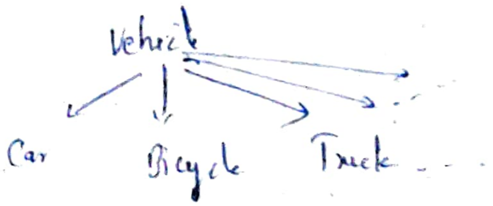


2) Multilevel inheritance

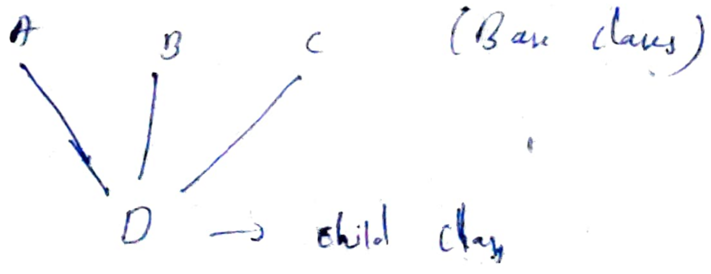


3) Hierarchical Inheritance

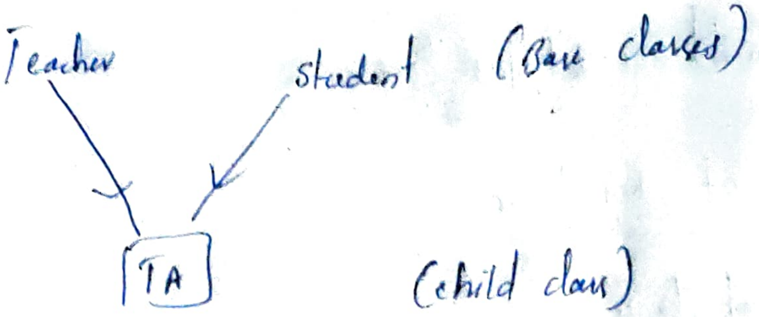




4) Multiple Inheritance



Ex:



ii.

Teacher.cpp

```
Class Teacher {
      public:
         string name;
         int age;
      void print () {
           cout << "Teacher" << endl;
      }
};
```

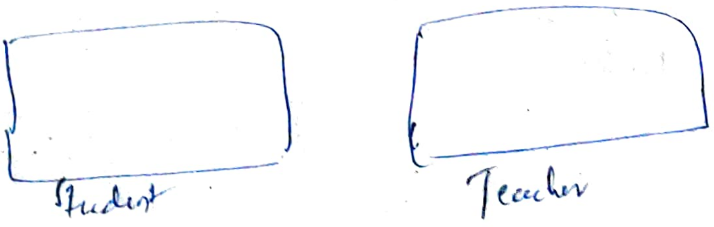
Student.cpp

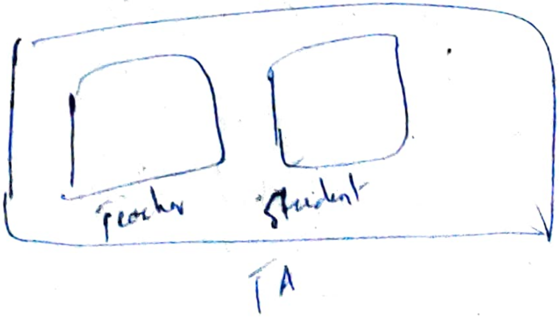
```
class student {
    public:
       ~~void print () {~~
     void print () {
      cout << "student" << endl
     }
};
```

```
class TA : public Teacher, Public student {



};
```





→ when we call TA, which constructor is called / invoked first?

Since we have written Teacher as first class so the constructor is invoked first

class TA : public Teacher , public student

↓ Constructor

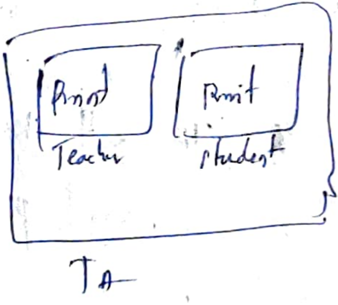
```
Teacher ()
Student ()
TA () .
```

⇒

```
int main () {
    TA a;
    a. print () .
```

↓

Q) which class print is called? →



we get error, If we dont specify the class name.

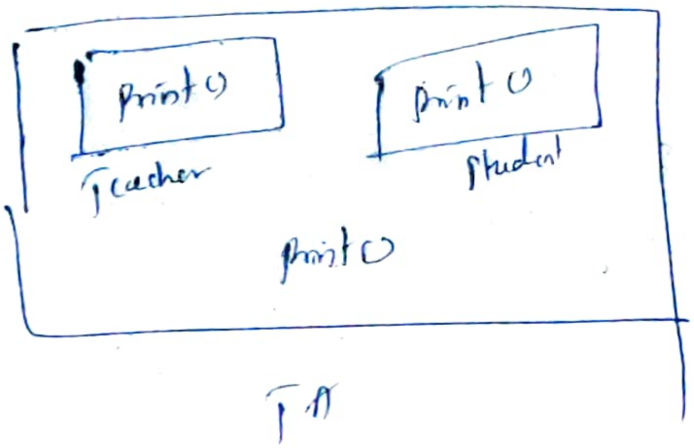
* The line can be modified as

```
a.student :: print()
```

what if

our, TA class has print() function

```
class TA : public teacher , public Student {
        public :
            void print() {
                cout << "TA"<<endl;
            }
};
```



- a.print() → doesn't gives an error now because
  ↓
- This will check, whether our TA class has any function named print() . If yes invoke,
  else it checks the parent class.